

**AUTOMATIC CONTROL METHOD FOR TRANSACTION QUANTITY**

Patent Number: JP6266669  
Publication date: 1994-09-22  
Inventor(s): MAENAMI TETSURO  
Applicant(s): HITACHI LTD  
Requested Patent: ☐ JP6266669  
Application Number: JP19930055456 19930316  
Priority Number(s):  
IPC Classification: G06F15/00; G06F13/00  
EC Classification:  
Equivalents:

---

**Abstract**

---

**PURPOSE:** To avoid the overload state of CPU and to prevent extreme response delay owing to the excess of CPU processing capacity.

**CONSTITUTION:** Transactions inputted from a transaction input/output part 10 are counted as the number of retained transactions in a system before they are distributed to an online job processing 30 in a transaction quantity automatic control part 70. When a CPU use rate noticed from a CPU use rate monitor part 20 exceeds the danger value of the CPU use rate which is previously set at every prescribed time, the number of retained transactions in the system per unit time tends to increase and the CPU use rate is judged to become 100% at next processing time, the input restriction of the new transaction is indicated to the transaction input/output part 10.

---

Data supplied from the esp@cenet database - I2

(19)日本国特許庁 (J P)

(12)公開特許公報 (A)

(11)特許出願公開番号

特開平6-266669

(43)公開日 平成6年 (1994) 9月22日

(51) Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/00	3 2 0 A	7459-5L		
13/00	3 5 1 Z	7368-5B		

審査請求 未請求 請求項の数5 O L (全 9 頁)

(21)出願番号 特願平5-55456

(22)出願日 平成5年 (1993) 3月16日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 前波 哲朗

東京都千代田区九段南二丁目3番14番 株式会社日立製作所公共情報事業部内

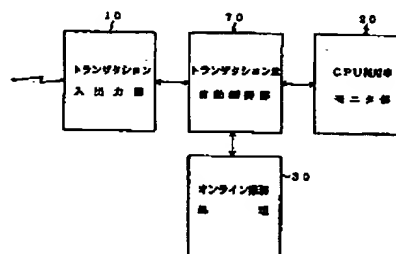
(74)代理人 弁理士 鈴木 誠

(54)【発明の名称】 トランザクション量自動制御方法

(57)【要約】

【目的】 CPUの過負荷状態を回避しCPU処理能力オーバーによる極端なレスポンス遅延を防止する。

【構成】 トランザクション量自動制御部70において、トランザクション入出力部10より入力したトランザクションをオンライン業務処理30に振り分ける前に、系内滞留トランザクション件数としてカウントする。次に、一定時間毎に、CPU利用率モニタ部20より通知されるCPU利用率が予め設定したCPU利用率の危険値を上回り、かつ、単位時間当りの系内滞留トランザクション件数が増加傾向にあり、次回処理時にCPU利用率が100%となると判断されるときに、トランザクション入出力部10に対し新規トランザクションの入力規制を指示する。



## 【特許請求の範囲】

【請求項1】 オンラインシステムにおいて、CPU利用率及び単位時間あたりの系内滞留トランザクション件数の変動量に基づいて、CPUの過負荷状態を予測して一時的に新規のトランザクションの入力を規制することを特徴とするトランザクション量自動制御方法。

【請求項2】 CPU利用率が危険値以上で、かつ、単位時間あたりの系内滞留トランザクション件数の変動量が増加傾向にある場合に限って、新規のトランザクションの入力を規制することを特徴とする請求項1記載のトランザクション量自動制御方法。

【請求項3】 オンラインシステムにおいて、単位時間あたりの系内滞留トランザクション件数及びその変動量に基づいて、CPUの過負荷状態を予測して一時的に新規のトランザクションの入力を規制することを特徴とするトランザクション量自動制御方法。

【請求項4】 単位時間あたりの系内滞留トランザクション件数が危険値以上で、かつ、単位時間あたりの系内滞留トランザクション件数の変動量が増加傾向にある場合に限って、新規のトランザクションの入力を規制することを特徴とする請求項3記載のトランザクション量自動制御方法。

【請求項5】 単位時間あたりの系内滞留トランザクション件数の危険値を、オンラインシステムの稼働統計情報に基づいて自動的に決定することを特徴とする請求項4記載のトランザクション量自動制御方法。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、オンラインシステムに係り、特に、入力するトランザクション量をCPU能力を合わせ自動制御するための方法に関する。

## 【0002】

【従来の技術】 従来、オンラインシステムにおけるトランザクション量の自動制御の方法として、次のような方式が知られている。

【0003】 (1) オンライン業務プログラムの処理時間もしくはCPU時間をタイマ監視し、当該タイマの設定時間を超える場合、トランザクション処理を一時中断し、制御をトランザクション起動側に戻すか、または、トランザクション処理自体を中断しCPUの過負荷状態を回避する。

【0004】 (2) オンラインシステム構築時に、設計段階で机上計算によりサポート可能な最大回線数、タスク数を算出するか、もしくは、テストにより取得した情報をもとにサポート可能な最大回線数、タスク数を算出しシステム構成を決定することにより、CPUの過負荷状態を回避する。

【0005】 なお、CPU処理能力を向上させる方式として、特開昭61-55722の記載のように、タイマ等の処理を専用に実行させることによりCPUの負荷を

軽減させる処理方式が知られている。

## 【0006】

【発明が解決しようとする課題】 しかし、前記(1)の方式では、トランザクション処理を一時中断するか完全に中断してしまうため、中断される前の処理が無駄になるか、トランザクション起動側との対話処理が煩雑となる。なお、公衆回線を使用するようなオンライン処理においては、回線使用料が過分に課金されるか、完全に無駄になる。

10 【0007】 また、前記(2)の方式では、人為的な計算・設定の誤り、もしくは、入力されるトランザクション種別の分布により、CPU処理能力が100%を超える可能性があり、極端なレスポンス遅延を起こす可能性があった。

【0008】 本発明の目的は、オンラインシステムにおいて前記諸問題点を解消できる新しいトランザクション量自動制御方法を提供することにある。

## 【0009】

【課題を解決するための手段】 本発明の骨子は、オンラインシステムにおいて、CPU利用率及び単位時間あたりの系内滞留トランザクション件数の変動量に基づき、あるいは単位時間あたりの系内滞留トランザクション件数及びその変動量に基づき、CPUの過負荷状態を予測して一時的に新規のトランザクションの入力を規制することにより、トランザクション量を自動的に制御することである。

## 【0010】

30 【作用】 システム稼働状況に応じ、CPUの過負荷状態が予測される時にトランザクションの新規入力を一時的に規制することによって、トランザクション処理を中断せずに、CPU処理能力オーバーによる極端なレスポンス遅延を回避することができ。

## 【0011】

【実施例】 図1は本発明の第1の実施例に係るオンラインシステムの概略ブロック図である。図1において、10は、オンラインシステムにおけるトランザクション入出力部であり、通信回線により接続された端末や他ホストとの間のトランザクションの入出力を制御する。20は、オンラインシステムの稼働中にCPU利用率を測定するモニタ部である。30は、オンラインシステムの業務処理を実行するオンライン業務処理部である。

40 【0012】 70は、本発明の方法によってトランザクション量の自動制御を行なうトランザクション量自動制御部である。これは、本実施例においては、単位時間あたりの入出力トランザクション件数を監視し、CPU利用率モニタ部20より通知されるCPU利用率が所定の値を超える場合に、系内滞留トランザクション件数の単位時間に関する変動量が増加傾向にあるかを計算し、増加傾向にある場合、CPUの過負荷状態を予測したとき  
50 らトランザクション入出力部10に対し新規トランザク

ションの入力規制を指示する。

【0013】次に図2乃至図4によって、トランザクション量自動制御部70と各部の動作を説明する。図2はトランザクション量自動制御部70における系内滞留トランザクション件数を管理する処理手順を示すフローチャート、図3はトランザクション量のカウンテーブルの説明図、図4はトランザクション量自動制御部70が所定時間間隔で実行する流量制御処理の手順を示すフローチャートである。

【0014】まず図2に関連して説明する。トランザクション入出力部10は、入力したトランザクションをオンライン業務処理部30に振り分ける前に、入力したトランザクション毎にトランザクション量自動制御部70に制御を渡す(100)。トランザクション量自動制御部70はトランザクション種別毎に入力件数をカウントアップし(102)、トランザクションカウンテーブル(図3)の該当トランザクション種別114の入力件数エリア116を更新し(104)、しかる後にオンライン業務処理部30へ制御を渡す(106)。

【0015】オンライン業務処理部30にて必要な処理を終了すると、トランザクション量自動制御部70に制御が渡る。トランザクション量自動制御部70は、トランザクション種別毎の出力件数をカウントアップし(108)、カウンテーブル(図3)の該当トランザクション種別114の出力件数エリア118を更新し(110)、トランザクション入出力部10に制御を渡す(112)。

【0016】次に図4に関連して説明する。オンラインシステム起動後、システムに予め設定された所定の時間間隔で当該処理がタイマ起動される(200)。まずトランザクション量自動制御部70は、システムに予め設定されたCPU利用率の危険値と、CPU利用率モニタ部20より取得した稼働中のCPU利用率を比較する(202)。

【0017】稼働中CPU利用率が危険値を上回っている場合、トランザクション量カウンテーブル(図3)の入力件数エリア116の値と出力件数エリア118の値より、単位時間あたりの系内滞留トランザクション件数( $N_t$ )を算出し、この値と前回処理起動時に計算した単位時間あたりの系内滞留トランザクション件数( $N_0$ )とから、前回処理起動時から今回処理起動時までの系内滞留トランザクション件数の変動量( $N_t - N_0$ )を計算する(204)。そして、この変動量が増加傾向( $N_t > N_0$ )に有るかチェックし(206)、増加傾向にない場合には当該処理を終了する(222)。

【0018】増加傾向にある場合には、CPU利用率が100%になる時刻を計算し(208)、次の処理起動迄にCPU利用率が100%を超えるかチェックする(210)。100%を超えない場合、当該処理を終了する(222)。

【0019】100%を超える場合には、トランザクション入出力部10へ入力規制を指示し(112)、入力規制フラグをONにし(214)、当該処理を終了する(222)。トランザクション入出力部10は、入力規制を指示された場合には、それが解除されるまで、トランザクションの新規入力を禁止する。

【0020】一方、前述のCPU利用率(危険値)とCPU利用率モニタ部20より取得した稼働中のCPU利用率の比較(202)で、稼働中CPU利用率が危険値を下回っていると判断した場合、入力規制フラグの状態をチェックする(216)。当該フラグがOFFの場合、当該処理を終了する(222)。

【0021】入力規制フラグがON状態の場合には、入力規制の解除をトランザクション入出力部10へ指示し(218)、入力規制フラグをOFFにし(220)、当該処理を終了する(222)。

【0022】以上のように、本実施例によれば、CPUの過負荷状態になる危険性がある期間だけ新規トランザクションの入力を一時的に規制することができるので、CPU処理能力オーバーによる極端なレスポンス遅延を回避することができる。そして、トランザクションの新規入力を規制する状況においても、トランザクション処理そのものを中絶するものではないので、それまでの処理が無駄になることはない。また、トランザクション起動側との面倒な対話処理も不要である。

【0023】また、稼働中CPU利用率が危険値を上回っていても、系内トランザクション件数が減少傾向にあれば、必ずしもCPUが過負荷状態になるとは限らない。そこで本実施例では、利用率が危険値に到達単位時間当りの系内トランザクション件数の変動量を計算し、それが増加傾向にあるか否かを判定する。そして、CPU利用率が危険値を上回っていても、系内トランザクション件数が減少傾向にあれば新規トランザクションの入力規制をせず、増加傾向の場合に、CPU利用率が100%になる迄の時間を計算し、次回監視時刻までにCPU利用率が100%となると判定した場合にはじめて新規トランザクションの入力を規制する。このような方式によれば、CPU利用率が危険値に達した時に即座に入力規制を行なう方式に比べ、CPU処理能力を最大限活用できる。

【0024】図5は本発明の第2の実施例に係るオンラインシステムの概略ブロック図である。図5において、トランザクション入出力部10、オンライン業務処理部30は図1中の対応部と同一である。トランザクション量自動制御部71は、図1中のトランザクション量自動制御部70と同様に、図2に示した手順によってトランザクション量カウンテーブル(図3)の更新を行なうが、タイマ起動によって定期的に行う流量制御処理が図6に示すような内容でトランザクション量自動制御部70のそれとは相違する。

【0025】稼働統計データ記憶部40は、OSもしくは、オンラインを制御するプログラムにより取得されるオンラインシステムのCPU利用率、トランザクション実行時間、トランザクション件数等の稼働統計情報を記憶する部分である。特性データ作成／更新部50は、稼働統計データ記憶部40内の情報をもとに、単位時間あたりのトランザクション種別毎のCPU利用率を計算し、その値により重み付けを行い正規化した単位時間あたりのトランザクション件数を計算し、その値と単位時間あたりのCPU利用率合計値より特性関数データを作成／更新する。トランザクション量自動制御に直接必要とされる特性関数データとしては、ここではCPU100%利用率に対応したトランザクション件数と、CPU利用率の危険値に対応したトランザクション件数が求められる。60は、このような特性関数データの記憶部である。

【0026】次に、トランザクション量自動制御部71による流量制御について説明する。図6は、その制御処理の手順を示すフローチャートである。

【0027】オンラインシステム起動後、特性関数データ記憶部60よりCPU利用率の危険値に対応するトランザクション件数とCPU100%利用率に対応するトランザクション件数を入力し、システムに予め設定された所定の時間間隔で処理がタイマ起動される(300)。

【0028】トランザクション量自動制御部71はまず、トランザクション量カウントテーブル(図3)の入力件数エリア116の値と出力件数エリア118の値より、単位時間あたりの系内滞留トランザクション件数( $N_1$ )を算出する(302)。次に、CPU利用率の危険値に対応するトランザクション件数を、単位時間あたりの系内滞留トランザクション件数( $N_1$ )と比較する(304)。

【0029】単位時間あたりの系内滞留トランザクション件数( $N_1$ )が、CPU利用率の危険値に対応するトランザクション件数を上回る場合、単位時間あたりの系内滞留トランザクション件数( $N_1$ )と前回処理起動時に計算した単位時間あたりの系内滞留トランザクション件数( $N_0$ )とより、系内トランザクション件数の変動量( $N_1 - N_0$ )を計算する(306)。そして、この変動量が増加傾向( $N_1 > N_0$ )に有るかチェックする(308)。増加傾向にない場合当該処理を終了する(324)。

【0030】増加傾向に有る場合、系内滞留トランザクション件数が、CPU100%利用率に対応した系内滞留トランザクション件数となる時刻を計算し(310)、次の当該処理起動迄にCPU利用率が100%を超える系内滞留トランザクション件数になるかチェックする(312)。CPU利用率が100%を超える系内滞留トランザクション件数にならない場合、当該処理

を終了する(324)。

【0031】CPU利用率が100%を超える系内滞留トランザクション件数になる場合、入力規制をトランザクション入出力部10へ指示し(314)、入力規制フラグをONにして(316)、当該処理を終了する(324)。

【0032】一方、CPU利用率の危険値に対応する系内滞留トランザクション件数と単位時間あたりの系内滞留トランザクション件数( $N_1$ )との比較(304)において、単位時間あたりの系内滞留トランザクション件数( $N_1$ )が、CPU利用率の危険値に対応する系内滞留トランザクション件数を下回っている場合、入力規制フラグの状態をチェックし(318)、当該フラグがOFFの場合は当該処理を終了する(324)。入力規制フラグがON状態であれば、入力規制の解除をトランザクション入出力部10へ指示し(320)、入力規制フラグをOFFにし(322)当該処理を終了する(324)。

【0033】以上のように、本実施例によれば、前記第1実施例と同様に、トランザクション処理を中断することなく、CPUの過負荷状態を回避し、極端なレスポンス遅延を防止できる。また、単位時間あたりの系内滞留トランザクション件数が危険値を上回っていても、系内滞留トランザクション件数が減少傾向にあれば新規トランザクションの入力規制を行わず、増加傾向にある場合にCPU100%利用率に対応した系内滞留トランザクション件数になる迄の時間を計算し、次回監視時刻までに当該系内滞留トランザクション件数となると判定される場合に新規トランザクションの入力を規制するので、CPU処理能力を有効に活用することができる。

【0034】更に、本実施例によれば、CPU利用率に対応した系内滞留トランザクション件数にて制御を行うのでCPU利用率をモニタする必要がなく、その分だけCPU負荷を軽減できるため、CPU処理能力をより効率的に利用できる。

【0035】

【発明の効果】以上に述べたように、本発明によれば以下のような効果を達成できる。

【0036】(1) CPU処理能力オーバーによる極端なレスポンス遅延を確実に回避することができ、

【0037】(2) 新規のトランザクションの入力を規制する方法であるため、トランザクション処理を中断する方法のような無駄がない。

【0038】(3) システム稼働状況に応じて新規トランザクション入力規制を行なうので、人為的な計算・設定に依存する方法のような人為的な誤りを回避でき、また机上計算・実測工数を削減できる。

【0039】(4) 単位時間あたりの系内滞留トランザクション件数の変動量が増加傾向にあることを考慮することによって、真にCPUの過負荷状態の危険がある状

況においてのみ新規のトランザクションの入力を規制し、CPU処理能力等のシステム資源を最大限に活用できる。

【図面の簡単な説明】

【図1】第1の実施例に係るオンラインシステムの概略ブロック図である。

【図2】トランザクション入出力件数管理処理を示すフローチャートである。

【図3】トランザクション量カウントテーブルの説明図である。

【図4】第1の実施例における流量制御処理を示すフローチャートである。

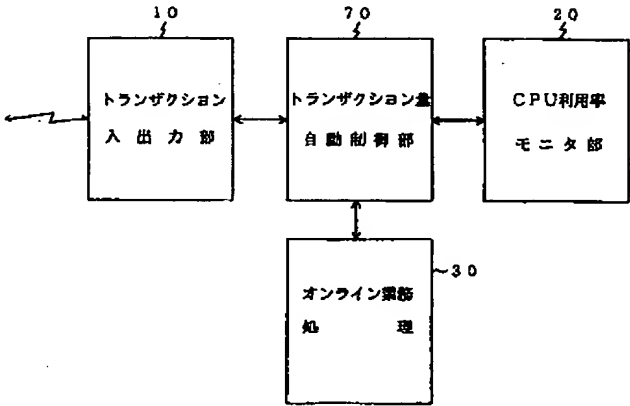
【図5】第2の実施例に係るオンラインシステムの概略

ブロック図である。  
【図6】第2の実施例における流量制御処理を示すフローチャートである。

【符号の説明】

- 10 トランザクション入出力部
- 20 CPU利用率モニタ部
- 30 オンライン業務処理部
- 40 稼働統計データ記憶部
- 50 特性関数データ作成／更新部
- 60 特性関数データ記憶部
- 70 トランザクション量自動制御部
- 71 トランザクション量自動制御部

【図1】



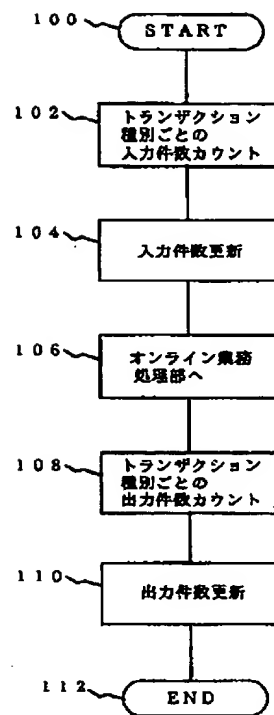
【図3】

トランザクション量カウントテーブル

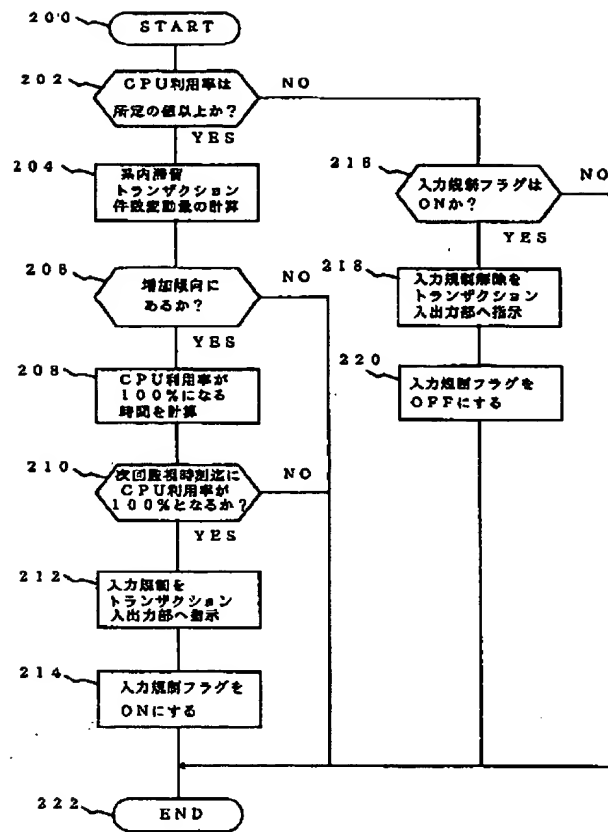
トランザクション種別	入力件数	出力件数

114 116 118

【図2】

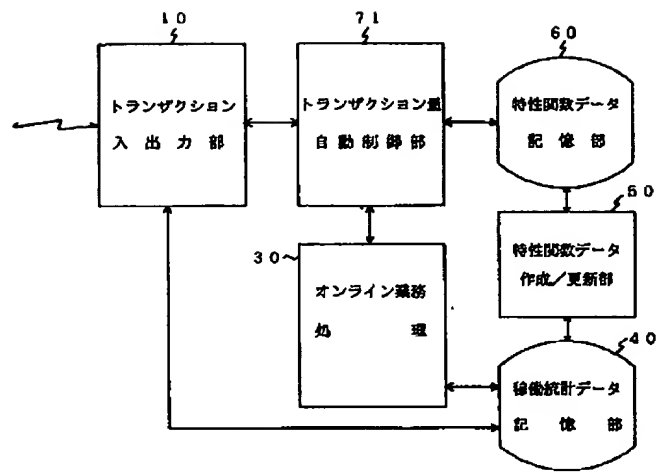


【図4】





【図5】



[図6]

